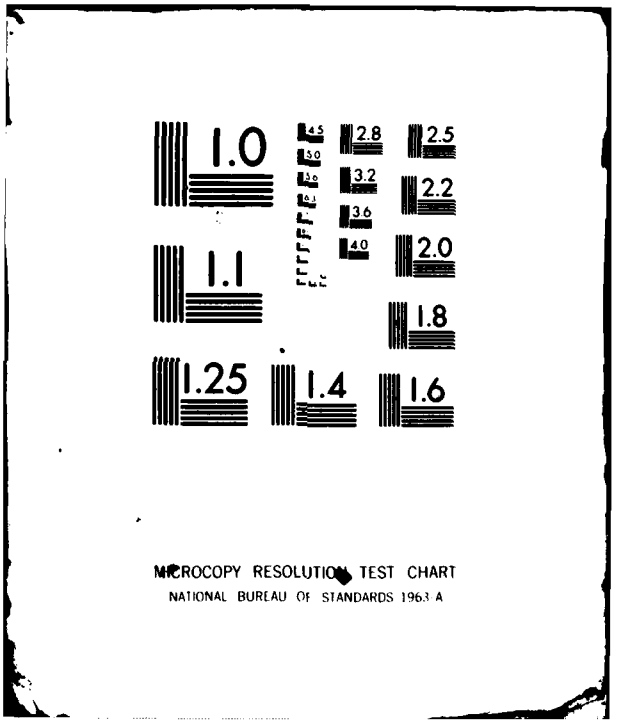
AD-A087 428 STANFORD UNIV CA SYSTEMS OPTIMIZATION LAB F/G 12/1
COMPUTING FINITE-DIFFERENCE APPROXIMATIONS TO DERIVATIVES FOR N--ETC(U)
MAY 80 P E GILL, W MURRAY, M A SAUNDERS N00014-75-C-0267
UNCLASSIFIED SOL-80-6 NL

END
DATE FILMED
9 80
DTIC


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

ARO 16470.11-M

Systems
Optimization
Laboratory

LEVEL II

ADA087428

DTIC
ELECTE
AUG 4 1980
S D
D

DDC FILE COPY

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

Department of Operations Research
Stanford University
Stanford, CA 94305

80 8 1 061

(12)

SYSTEMS OPTIMIZATION LABORATORY
DEPARTMENT OF OPERATIONS RESEARCH
STANFORD UNIVERSITY
STANFORD, CALIFORNIA 94305

(12) 27

(6) COMPUTING FINITE-DIFFERENCE APPROXIMATIONS TO DERIVATIVES FOR NUMERICAL OPTIMIZATION.

by

(10) Philip E. Gill, Walter Murray,
Michael A. Saunders and Margaret H. Wright

(9) TECHNICAL REPORT SOL 80-6
(11) May 1980

(14) SOL-80-6

Research and reproduction of this report were supported by the Department of Energy Contract DE-AC03-76SF00326, PA No. DE-AT03-76ER72018; National Science Foundation Grants MCS-7926009 and ENG77-06761; the Office of Naval Research Contract (15) N00014-75-C-0267, and the U.S. Army Research Office Contract DAAG29-79-C-0110.

408765

Computing Finite-Difference Approximations To Derivatives For Numerical Optimization

Philip E. Gill, Walter Murray, Michael A. Saunders and Margaret H. Wright
Systems Optimization Laboratory
Department of Operations Research
Stanford University
Stanford, California 94305

ABSTRACT

Finite-difference approximations to derivatives are useful not only in optimization algorithms, but also in other circumstances such as sensitivity analysis. In this paper we discuss methods for estimating the relative cancellation error and relative truncation error in a finite-difference approximation and propose a technique for computing the finite-difference interval so that the bounds upon the errors are balanced. We also propose a method for choosing the finite-difference interval in a quasi-Newton algorithm for unconstrained minimization that uses function values only.

1. Introduction

Let $F(x)$, $x \in \mathbb{R}^n$, be a twice continuously differentiable nonlinear function. If the derivatives of $F(x)$ are too difficult or expensive to evaluate, it is now generally accepted that the best methods available for minimising $F(x)$ when n is not too large are based on using quasi-Newton methods with finite-difference approximations to the gradient vector. However, the success of such algorithms critically depends on obtaining "good" approximations to the necessary first derivatives — much more so than with Newton-type methods that use finite differences of gradients to approximate second derivatives. As we shall see, certain standard choices for the finite-difference interval produce acceptable approximations for well scaled problems. However, they may be disastrous in the presence of bad scaling or a non-typical starting point. Although no procedure is guaranteed for every case, the methods suggested in this paper are designed to overcome the most common difficulties in choosing the finite-difference interval, and can lead to substantial improvements in performance of the associated minimisation algorithms.

Finite-difference approximations to the gradient vector are derived from expanding F in a Taylor series along an appropriate vector, i.e.:

$$F(x + h_j e_j) = F(x) + h_j g_j(x) + \frac{1}{2} h_j^2 G_{jj}(x) + O(h_j^3), \tag{1}$$

where h_j is the finite-difference interval corresponding to the j-th variable, e_j is the j-th column of the identity matrix, g_j is the j-th component of the gradient vector $g(x)$, and G_{jj} is the j-th diagonal element of the Hessian matrix $G(x)$. The two most common finite-difference formulae are:

(i) the *forward-difference formula*

$$g_j \approx \frac{1}{h_j}\big(F(x + h_j e_j) - F(x)\big); \tag{2}$$

(ii) the *central-difference formula*

$$g_j \approx \frac{1}{2h_j}\big(F(x + h_j e_j) - F(x - h_j e_j)\big), \tag{3}$$

where $h_j = \delta_j(1 + |x_j|)$. (Note that h_j is an *absolute* perturbation and that δ_j is a *relative* perturbation. When $0 \leq |x_j| \leq 1$, they are essentially equivalent.)

The *truncation error* in (2) — i.e., the error due to neglecting terms of the series (1) — is $O(h_j)$, and the truncation error associated with formula (3) is $O(h_j^2)$. The higher order error of the central-difference formula is obtained at the cost of an additional function evaluation for each component of the gradient.

The main consideration in the implementation of finite-difference techniques is the choice of the n values $\{h_j\}$. Since the truncation error in either (2) or (3) decreases with h_j, it might appear that choosing the finite-difference interval as small as possible will minimise the error. However, as h_j becomes smaller, the function values in (2) and (3) become closer and, as we shall see, computational error becomes increasingly important. In this paper we consider techniques for choosing h_j effectively that take into account both sources of error.

We shall assume that all computation is carried out on a machine that stores a non-zero representable number $\hat{f}$ in the form

$$\hat{f} = m\beta^e, \tag{4}$$

where β is the machine base, e is the exponent and m is a mantissa comprising r digits. All floating-point numbers are normalised such that

$$\frac{1}{\beta} \leq |m| < 1.$$

It is convenient (though not essential) to assume that the machine performs *correct rounding*, that is, if $\hat{f}$ denotes the floating-point value of a non-zero number f, then

$$\left|\frac{\hat{f} - f}{f}\right| \leq \frac{1}{2}\beta^{1-r}.$$

The number β^{1-r} is termed the *relative machine precision* and will be denoted by ϵ. For a complete discussion of the errors in floating-point arithmetic, see Wilkinson (1963).

Throughout this paper, it will be important to distinguish "exact" from "computed" quantities. Let γ denote an exact quantity, and let $\hat{\gamma}$ denote an approximation that satisfies $\hat{\gamma} = (1 + \sigma)\gamma$; the value $|\sigma|$ will be termed the *relative error* in the approximation.

2. Estimating the cancellation error

Numbers computed in finite precision necessarily deviate from the corresponding "exact" quantities. Fortunately, each occurrence of rounding or floating-point arithmetic typically introduces a relative error bounded by the order of machine precision, and in most instances this level of error, even accumulated over many operations, is "negligible". However, certain computations carry the risk of introducing much greater relative error — in particular, the subtraction of nearly equal rounded numbers. The error associated with this procedure is

termed *cancellation error* (see Kahan, 1973, for a detailed treatment of related topics).

Consider two numbers f_1 and f_2, whose floating-point values are $\hat{f}_1 = f_1(1+\epsilon_1)$ and $\hat{f}_2 = f_2(1+\epsilon_2)$, where ϵ_1 and ϵ_2 are bounded in magnitude by the relative machine precision. The exact difference of $\hat{f}_1$ and $\hat{f}_2$ can be written as:

$$\Delta\hat{f} \equiv f_1(1+\epsilon_1) - f_2(1+\epsilon_2) = (f_1 - f_2)(1+\eta), \tag{5}$$

so that η represents the relative error in $\Delta\hat{f}$ with respect to the exact difference of the original numbers.

If $f_1 = f_2$, we say that *complete cancellation* occurs. Otherwise, re-arranging (5) gives an expression for η:

$$\eta = \frac{\epsilon_1 f_1 - \epsilon_2 f_2}{f_1 - f_2}. \tag{6}$$

The relative error in $\Delta\hat{f}$ may therefore be bounded as follows:

$$\begin{aligned} |\eta| &= \left|\frac{\epsilon_1 f_1 - \epsilon_2 f_2}{f_1 - f_2}\right| \\ &= \left|\frac{\epsilon_2(f_1 - f_2) + f_1(\epsilon_1 - \epsilon_2)}{f_1 - f_2}\right| \\ &\leq \epsilon\left(1 + 2\left|\frac{f_1}{f_1 - f_2}\right|\right), \end{aligned} \tag{7}$$

assuming that $|f_1| \geq |f_2|$.

If $|f_1 - f_2|$ is small relative to $|f_1|$ (i.e., f_1 "nearly equals" f_2), (7) shows that the relative error in $\Delta\hat{f}$ is not restricted to be of order ϵ. The error may be large *not* because of errors in subtracting $\hat{f}_1$ and $\hat{f}_2$ (since $\Delta\hat{f}$ is their *exact* difference), but rather because of the initial errors incurred in rounding f_1 and f_2; note that if ϵ_1 and ϵ_2 are zero, $\eta = 0$. If f_1 nearly equals f_2, the original high-order significant digits cancel during subtraction, which means that low-order digits discarded in rounding are the most significant digits of the exact result — i.e., cancellation reveals the error of rounding. If f_1 and f_2 are not similar, the bound on the cancellation error becomes of the same order as the error resulting from any other floating-point operation, and is not of any special significance.

As an example, consider the subtraction of the numbers

$$f_1 = .2946796847 \quad \text{and} \quad f_2 = .2946782596 \tag{8}$$

on a machine with a mantissa of six decimal digits ($\epsilon = 10^{-5}$). If correct rounding is used, the values of $\hat{f}_1$ and $\hat{f}_2$ are .294680 and .294678, respectively, with

the difference $\Delta\hat{f}$ (computed exactly) being $.2 \times 10^{-5}$. However, the difference between the exact values of f_1 and f_2 is $.14251 \times 10^{-5}$, which implies that $\Delta\hat{f}$ has a relative cancellation error of $\eta = .40341$, computed from (6). From (7) it can be seen that the bound on relative cancellation error decreases with ϵ; if eight figures are used to represent f_1 and f_2, the relative cancellation error is $.357 \times 10^{-2}$.

It is not possible to compute the exact value (6) of the relative cancellation error without utilising the exact values of f_1 and f_2, and exact arithmetic. Therefore, we can only estimate the bound (7) on the cancellation error. For convenience, we shall usually refer to the estimate of a bound on the cancellation error as simply the "cancellation error", and shall be concerned with *computable* estimates of such a bound.

One possible estimate is based upon finding the difference in the mantissas of $\hat{f}_1$ and $\hat{f}_2$. Suppose that $\hat{f}_1 = m_1\beta^e$, where m_1 is the mantissa of $\hat{f}_1$ and e its exponent. The number $\hat{f}_2$ may be written in the (possibly unnormalized) form $\hat{f}_2 = m_2\beta^e$. A measure of the relative cancellation error is given by

$$\hat{\eta} = \begin{cases} \beta^{-r}/|m_1 - m_2|, & m_1 \neq m_2, \\ 1, & m_1 = m_2. \end{cases} \tag{9}$$

If this formula is used for the example (8), we obtain $\hat{\eta} = .5$ on a six-digit decimal machine and $\hat{\eta} = .704 \times 10^{-2}$ on an eight-digit machine.

Note that we adopt the convention that $\hat{\eta}$ is unity when complete cancellation takes place. A finite upper bound on the estimated cancellation error is essential in order to compute $\hat{\eta}$, but implies that $\hat{\eta}$ will underestimate η if m_1 is very close to m_2.

Formula (9) is related to a good "rule of thumb" method in which the relative cancellation error is estimated by

$$\hat{\eta} = \epsilon\beta^k, \tag{10}$$

where k is the number of leading coincident digits in m_1 and m_2.

A serious disadvantage of (9) and (10) is that it is not a straightforward process to obtain the mantissa of a floating-point number when using a high-level language such as Fortran. An alternative formula that uses the floating-point values of $\hat{f}_1$ and $\hat{f}_2$ may be derived as follows. Assume that $\hat{f}_1$ and $\hat{f}_2$ are such that $|\hat{f}_1| \geq |\hat{f}_2|$. Multiplying the denominator and numerator of (9) by β^e we obtain

$$\frac{\beta^{-r}}{|m_1 - m_2|} = \frac{\beta^e\beta^{-r}}{|m_1\beta^e - m_2\beta^e|} = \frac{\beta^e\beta^{1-r}}{\beta|\hat{f}_1 - \hat{f}_2|}.$$

Because $|m_1| \geq 1/\beta$, we can write

$$\frac{\beta^c \beta^{1-r}}{\beta|\hat{f}_1 - \hat{f}_2|} \leq \beta^{1-r}\left|\frac{\beta^c m_1}{\hat{f}_1 - \hat{f}_2}\right|,$$

which leads to the estimate

$$\eta = \epsilon\left|\frac{\hat{f}_1}{\hat{f}_1 - \hat{f}_2}\right|. \tag{11}$$

Since this formula involves only ϵ and the floating-point values of f_1 and f_2, it can be easily implemented in a high-level language. However, we have incurred two penalties in exchange for this ease of use. Firstly, the cancellation error is overestimated by the amount that the mantissa of $\hat{f}_1$ is larger than $1/\beta$. The mantissa could be almost β times larger than $1/\beta$ and consequently, on average, (11) will be a better estimate of η on a binary machine than on any other. On a machine with a large value of β it may be worth computing the mantissas of $\hat{f}_1$ and $\hat{f}_2$ and using (9), but this must be done carefully to avoid introducing further rounding error in m_1 and m_2. One way of computing the mantissa of a floating-point number is to multiply it by powers of β until the product lies between $1/\beta$ and 1. However, this technique will be applicable only if the machine arithmetic can be relied upon to change the exponent without altering the mantissa when a number is multiplied by a power of the base. This method may also be too expensive in terms of the number of operations required. For example, later we shall require the relative cancellation error involved in computing the finite-difference approximation of each element of the gradient vector of a multivariate function. If the number of variables is large, the extra computation involved in finding the mantissas of $\hat{f}_1$ and $\hat{f}_2$ may be prohibitive.

The second disadvantage of (11) is that the resulting value of η will not tend monotonically to unity (the convention adopted to represent complete cancellation) as $\hat{f}_1$ approaches $\hat{f}_2$; for example, the value of η for the values of f_1 and f_2 given by (8) with six-digit precision ($\epsilon = 10^{-5}$) is 1.473. This problem is easily overcome by redefining η as

$$\eta = \frac{|\hat{f}_1|\epsilon}{|\hat{f}_1 - \hat{f}_2| + |\hat{f}_1|\epsilon}. \tag{12}$$

If (12) is used in example (8), we obtain $\eta = .5957$ for a six-digit mantissa and $\eta = .7028 \times 10^{-2}$ for an eight-digit mantissa.

It is important to note that during the derivation of all the preceding formulae for the relative cancellation error, we have assumed that the errors in both $\hat{f}_1$ and $\hat{f}_2$ are of the order of the machine precision. If this is not the case, the

formulae may not be appropriate. Suppose that the values (8) are computed on a four-digit machine and then represented on a six-digit machine, so that the last two figures are unreliable — for example, in this case it might happen that $\hat{f}_1 = .294750$ and $\hat{f}_2 = .294722$, so that the exact cancellation error in their computed difference is $.1864774 \times 10^2$. However, (12) gives $\eta = .9523 \times 10^{-1}$, a considerable underestimate of the error. The required estimate of cancellation error must involve only the *correct* digits of $\hat{f}_1$ and $\hat{f}_2$. Hence, if $\hat{f}_1$ and $\hat{f}_2$ are computed to a relative precision of σ, that is

$$\hat{f}_1 = f_1(1+\sigma_1) \quad \text{and} \quad \hat{f}_2 = f_2(1+\sigma_2),$$

where $|\sigma_1|, |\sigma_2| \leq \sigma$, then the formula

$$\eta = \frac{|\hat{f}_1|\sigma}{|\hat{f}_1 - \hat{f}_2| + |\hat{f}_1|\sigma} \tag{13}$$

should be used instead of (12). This distinction will be important when estimating the cancellation error associated with approximations to second derivatives.

3. Errors in finite-difference approximations

For simplicity of description, the discussion in the next two sections will concern estimating derivatives of the twice continuously differentiable univariate function $f(x)$, but all results apply directly to the multivariate case. We shall assume that f can be computed to full machine precision.

The first type of approximate derivative to be considered is the forward difference formula with interval h:

$$\varphi_p(h) = \frac{f(x+h) - f(x)}{h}. \tag{14}$$

We stress that φ_p is meant to denote the exact expression (14), obtained by applying exact arithmetic to the exact values of all quantities. The truncation error in the mathematical approximation (14) consists of the neglected terms in the Taylor series (1):

$$\varphi_p(h) - f'(x) = \frac{1}{2}hf''(x) + O(h^2). \tag{15}$$

Of course, $\varphi_p(h)$ is not available in the real world, and thus we shall be concerned with its computed version, whose evaluation involves several operations of rounding and floating-point arithmetic. Let $\hat{a}$ denote the computed version

of the exact quantity a, and define the intermediate computed quantity

$$\hat{\Delta}_r \equiv \hat{f}(\hat{x}+\hat{h}) - \hat{f}(\hat{x}), \tag{16}$$

where $+$ and $-$ in (16) denote *floating-point* addition and subtraction. The computed quantity $\hat{\phi}_r$ that corresponds to (14) is then given by

$$\hat{\phi}_r \equiv \hat{\Delta}_r / \hat{h},$$

where floating-point division is implied.

The previous discussion of errors in computation of quantities like $\hat{\Delta}_r$ shows that for "small" h, the relative error in $\hat{\phi}_r$ compared to φ_r will be dominated by the cancellation error arising from computing $\hat{\Delta}_r$, and henceforth we shall consider only this source of computational error in $\hat{\phi}_r$ to be significant. If the other errors from computation were included, this would only add a number of factors of the form $(1+\epsilon)$ to the error estimates.

Using (13), a computable estimate of the cancellation error in $\hat{\phi}_r$ is given by

$$\hat{\eta}_r = \frac{|\hat{f}(\hat{x})|\epsilon}{|\hat{f}(\hat{x}) - \hat{f}(\hat{x}+\hat{h})| + |\hat{f}(\hat{x})|\epsilon}, \tag{17}$$

where the arithmetic operations in (17) refer to floating-point operations, and for simplicity we have assumed that $|\hat{f}(\hat{x})| \geq |\hat{f}(\hat{x}+\hat{h})|$. We adopt the convention that $\hat{\eta}_r$ is unity if both $\hat{f}(\hat{x})$ and $\hat{f}(\hat{x}+\hat{h})$ vanish.

When second derivatives are to be approximated by finite differences, a similar analysis may be carried out for this case. Define $\Phi(h)$ as the exact quantity

$$\Phi(h) = \frac{1}{h^2}\big(f(x+h) - 2f(x) + f(x-h)\big), \tag{18}$$

and $\hat{\Phi}$ as its computed version.

As with $\hat{\phi}_r$, we can assume that the only significant source of computational error in $\hat{\Phi}$ is due to cancellation in subtracting nearly equal rounded function values. An important aspect of computing $\hat{\Phi}$ is the use of two intermediate values: $\hat{\phi}_r$ and the corresponding backward-difference value $\hat{\phi}_l$, defined as

$$\hat{\phi}_l \equiv \frac{\hat{f}(\hat{x}) - \hat{f}(\hat{x}-\hat{h})}{\hat{h}}, \tag{19}$$

which is computed using a value $\hat{\Delta}_l$ analogous to (16). Hence, $\hat{\Phi}$ can be written as

$$\hat{\Phi} = \frac{\hat{\phi}_r - \hat{\phi}_l}{\hat{h}}, \tag{20}$$

where the arithmetic operations are performed in floating-point.

Once again, it is reasonable to attribute the major error in computing $\hat{\Phi}$ to cancellation error; in this case, however, there is cancellation error not only in forming $\hat{\Phi}$, but also in the intermediate values $\hat{\varphi}_r$ and $\hat{\varphi}_s$. Formula (13) can be applied to (20) to obtain Υ, an estimate of the computational error in $\hat{\Phi}$:

$$\Upsilon = \frac{|\hat{\varphi}_r|\sigma}{|\hat{\varphi}_r - \hat{\varphi}_s| + |\hat{\varphi}_r|\sigma}, \tag{21}$$

where σ is an upper bound on the cancellation errors associated with $\hat{\varphi}_r$ and $\hat{\varphi}_s$. To compute σ, we estimate $\hat{\eta}_r$ from formula (17), and also the analogous quantity $\hat{\eta}_s$ corresponding to $\hat{\varphi}_s$; σ is then given by $\max(\hat{\eta}_r, \hat{\eta}_s)$. It is important to note that $\hat{\Phi}$ will be a reasonable estimate only if σ is sufficiently small.

4. Computing an estimate of the optimal finite-difference interval

4.1 Effects of h on cancellation and truncation errors.

Changes in the size of the finite-difference interval tend to have opposite effects on truncation and cancellation error. When the value $\hat{\varphi}_r$ is used as an approximation to f', for example, the truncation error is dominant for large values of h, since computation introduces only negligible error; as h decreases, truncation error in the exact value φ_r decreases, but the cancellation error in computing $\hat{\varphi}_r$ becomes larger.

To illustrate this phenomenon, consider the function

$$f(x) = (e^x - 1)^2 + \left(\frac{1}{\sqrt{1+x^2}} - 1\right)^2, \tag{22}$$

which has been evaluated for various values of h at the point $x = 1$, using short precision on an IBM 370. The smallest value of h that will register a change in x during floating-point addition is the machine precision $\epsilon = 16^{-5} \approx .95 \times 10^{-6}$. The function was computed with h values increasing from ϵ in multiples of ten. Table 1 contains the results of the computation. The first three columns contain the values of h, the computed function value at $x = 1$, and the computed function value at $x + h$. The fourth column contains values of $T(h)$, the relative truncation error incurred by using the exact value $\varphi_r(h)$ to approximate f'. The fifth column contains an estimate of the relative cancellation error in $\hat{\varphi}_r$, using formula (13) with $\sigma = \epsilon$. The final column contains the computed values of $\hat{\varphi}_r$. The exact value of $f'(x)$ (rounded to six figures) is $.954866 \times 10^1$.

Table 1

Cancellation and truncation errors in $\hat{\varphi}_F$ with $\epsilon = .953674 \times 10^{-6}$

h	$f(x)$	$f(x+h)$	$T(h)$	$\eta_F(h)$	$\hat{\varphi}_F$
ϵ	$.303828 \times 10^{1}$	$.303828 \times 10^{1}$	$.121183 \times 10^{-5}$	$.302670 \times 10^{0}$	$.700000 \times 10^{1}$
10ϵ	$.303828 \times 10^{1}$	$.303837 \times 10^{1}$	$.121180 \times 10^{-4}$	$.309916 \times 10^{-1}$	$.950000 \times 10^{1}$
$10^{2}\epsilon$	$.303828 \times 10^{1}$	$.303919 \times 10^{1}$	$.121188 \times 10^{-3}$	$.318226 \times 10^{-2}$	$.952000 \times 10^{1}$
$10^{3}\epsilon$	$.303828 \times 10^{1}$	$.304739 \times 10^{1}$	$.121264 \times 10^{-2}$	$.318730 \times 10^{-3}$	$.955800 \times 10^{1}$
$10^{4}\epsilon$	$.303828 \times 10^{1}$	$.313045 \times 10^{1}$	$.122027 \times 10^{-1}$	$.323888 \times 10^{-4}$	$.966490 \times 10^{1}$

The results indicate that for small h, the error in $\hat{\varphi}_F$ can be dominated by cancellation error to the extent that *no figures may be correct*. As the interval is increased, the relative truncation error increases, but the relative cancellation error decreases. The truncation error in approximating f' by $\hat{\varphi}_F$ is approximately linear in h and the cancellation error is approximately linear in $1/h$; this property will have important implications in methods for computing reasonable values of h. Clearly there is an optimal value of h that "balances" the relative truncation and cancellation errors, i.e. for which these errors are approximately equal. Examination of Table 1 indicates that the optimal value of h for the computed example lies between $h = .95 \times 10^{-4}$ and $h = .95 \times 10^{-3}$.

The relationship between cancellation errors in $\hat{\varphi}_F$, $\hat{\varphi}_B$, and $\hat{\Phi}$ is significant in procedures for estimating f' and f''. Note that Υ, the cancellation error in $\hat{\Phi}$, depends not only on the closeness of $\hat{\varphi}_F$ and $\hat{\varphi}_B$, but also on the value of σ, which reflects the cancellation error in computing the first-order estimates of f'. When h is so small that cancellation error dominates $\hat{\varphi}_F$ and $\hat{\varphi}_B$, the cancellation error in $\hat{\Phi}$ will also be large, not because $\hat{\varphi}_F$ and $\hat{\varphi}_B$ are nearly equal, but rather because σ is large. In addition, the value of Υ remains large even when h has been increased enough so that $\hat{\varphi}_F$ and $\hat{\varphi}_B$ are most accurate. The cancellation error in $\hat{\Phi}$ becomes reasonable only when two conditions are satisfied: h is large enough so that the cancellation errors $\hat{\eta}_F$ and $\hat{\eta}_B$ are small, and so that $\hat{\varphi}_F$ and $\hat{\varphi}_B$ differ by a significant amount.

In Table 2, we illustrate this relationship for the function (22). The exact value of f'', rounded to six figures, is $.242661 \times 10^{2}$; note that $\hat{\Phi}$ is wrong by several orders of magnitude until h has exceeded the "optimal" value for computing $\hat{\varphi}_F$ and $\hat{\varphi}_B$. This point is important in methods based on approximating both f' and f'' by finite differences.

4.2 Balancing truncation and cancellation errors.

To find an interval that balances truncation and cancellation errors, it is necessary to compute estimates of these quantities. If f' is not pathologically small

Table 2

Errors in approximating f' and f''

h	η_F	η_B	Υ	Φ
ϵ	$.302670 \times 10^{0}$	$.233028 \times 10^{0}$	$.502215 \times 10^{0}$	$-.314573 \times 10^{7}$
10ϵ	$.309916 \times 10^{-1}$	$.300706 \times 10^{-1}$	$.503060 \times 10^{0}$	$-.314572 \times 10^{5}$
$10^{2}\epsilon$	$.318226 \times 10^{-2}$	$.316805 \times 10^{-2}$	$.431999 \times 10^{0}$	$-.419430 \times 10^{3}$
$10^{3}\epsilon$	$.318730 \times 10^{-3}$	$.318477 \times 10^{-3}$	$.126600 \times 10^{0}$	$.220200 \times 10^{2}$
$10^{4}\epsilon$	$.323888 \times 10^{-4}$	$.322049 \times 10^{-4}$	$.135330 \times 10^{-2}$	$.242221 \times 10^{2}$
$10^{5}\epsilon$	$.376934 \times 10^{-5}$	$.358786 \times 10^{-5}$	$.175133 \times 10^{-4}$	$.243519 \times 10^{2}$

with respect to f and f'', and $h \gg \epsilon$, the cancellation error in $\hat{\varphi}_F$ (from (13)) is approximately a linear function of $1/h$. Hence, for two intervals h_1 and h_2,

$$\hat{\eta}_F(h_1) \approx \frac{h_2}{h_1}\hat{\eta}_F(h_2). \tag{23}$$

The truncation error clearly involves higher derivatives of f, which are (by assumption) not available. Therefore, the procedure for obtaining an "optimal" interval utilizes *estimates* of the needed values. In the remainder of this subsection, we shall assume that sufficiently accurate approximations are available; in Section 4.4 we present a procedure for obtaining such estimates using function values only.

We seek $\hat{T}(h)$, a computable estimate of the relative truncation error, that may be balanced against the relative cancellation error $\hat{\eta}_F$. The idea of balancing these two sources of error carries the implicit assumption that they are measured in a comparable manner. In particular, since the measure of relative cancellation error is bounded above by unity, this requirement is also imposed on $\hat{T}(h)$.

In almost all instances, a suitable model for $\hat{T}(h)$ is

$$\hat{T}(h) = \frac{h|f''|}{2|f'| + h|f''|}, \tag{24}$$

which is based on the first neglected term of the Taylor series. This expression for $\hat{T}(h)$ is bounded above by unity; we also adopt the convention that $\hat{T}(h) = 1$ when $f' = f'' = 0$.

If $|f''|$ is pathologically small with respect to $|f'|$, a more complicated expression for $\hat{T}(h)$ is appropriate:

$$\hat{T}(h) = \frac{h|f'' + \frac{1}{3}h(|f'''| + 1)|}{2|f'| + h|f'' + \frac{1}{3}h(|f'''| + 1)|}. \tag{25}$$

Let $\hat{h}$ denote an estimate of the absolute interval for which truncation and cancellation error are balanced. Assuming that $\hat{\eta}_r$ has been computed for some interval h, $\hat{h}$ can be obtained by equating (23) and (24):

$$\frac{h}{\hat{h}}\hat{\eta}_r(h) = \frac{\hat{h}|f''|}{2|f'| + \hat{h}|f''|}. \tag{26}$$

The desired $\hat{h}$ is the positive root of the resulting quadratic equation

$$\hat{h}^2|f''| - \hat{h}h|f''|\hat{\eta}_r(h) - 2h|f'|\hat{\eta}_r(h) = 0, \tag{27}$$

so that

$$\hat{h} = \frac{h}{2}\left(\hat{\eta}_r + \sqrt{\hat{\eta}_r^2 + \frac{8\hat{\eta}_r|f'|}{h|f''|}}\right). \tag{28}$$

When formula (25) must be used to estimate $\hat{T}(h)$, $\hat{h}$ is the solution of a cubic equation.

4.3 Finding intervals for well-scaled functions.

For many functions, a near-optimal value of h can be derived by inspection. Suppose that at a point x $(-1 \leq x \leq 1)$, the values of the function f and all its derivatives are of the same order of magnitude; the balancing relationship (26) then yields an $\hat{h}$ of order $\sqrt{\epsilon}$. This result can also be derived by inspection, as follows.

The relative truncation error in approximating f' by φ_r is $O(h)$; furthermore, a perturbation h induces a relative change of $O(h)$ in the function value and its derivatives. Figure 1 depicts the mantissas of $f(x)$, $f(x+h)$ and $\hat{\Delta}f = f(x+h) - f(x)$ on a six-digit machine, with $h = \sqrt{\epsilon}$, where the errors in rounding have been ignored. This perturbation causes half the significant figures of $f(x+h)$ to be different from those of $f(x)$; the identical digits are marked with a ".". In computing $\hat{\Delta}f$, only half the precision of f and $f(x+h)$ will be retained, and $\hat{\Delta}f$ itself will have a relative precision of only $\sqrt{\epsilon}$, so that the three least significant digits of $\hat{\Delta}f$, marked with a "×", are unreliable. Thus, the truncation and cancellation errors in $\hat{\varphi}_r$ are of similar order; it is this reasoning that underlies the "folklore" value of $\sqrt{\epsilon}$ as the "best" finite-difference interval.

mantissa of $f(x)$	·	·	·	m_4	m_5	m_6
mantissa of $f(x+h)$	·	·	·	m_4	m_5	m_6
mantissa of $\hat{\Delta}f$	m_1	m_2	m_3	×	×	×

Figure 1. Schematic diagram of six-digit mantissas during first-order differencing.

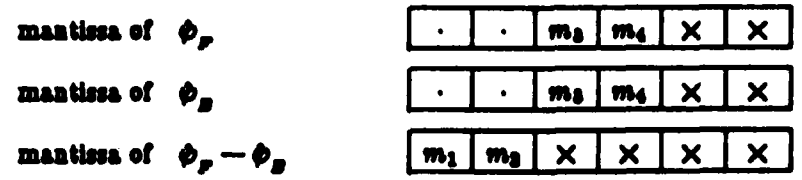


Figure 2. Schematic diagram of six-digit mantissas during second-order differencing.

Similar perturbation analyses can be used to estimate nearly optimal finite-difference intervals for other approximations. For the central-difference formula φ_c defined by

$$\varphi_c(h) = \frac{f(x+h) - f(x-h)}{2h}, \tag{29}$$

the truncation error is $O(h^2)$; however, the relative cancellation error in the computed version $\hat{\varphi}_c$ remains at $O(\epsilon/h)$ (estimated from (13), assuming that $h \gg \epsilon$). This implies that the optimal finite-difference interval is $O(\sqrt[3]{\epsilon})$.

Finding the optimal interval to approximate f'' using (18) is more complicated, as noted in Section 4.1. Let $\hat{\Phi}$ be computed by the formula (20). In this case, we wish to balance the truncation error and cancellation error as usual, but it is essential to include the effects of cancellation error in computing the intermediate quantities $\hat{\varphi}_p$ and $\hat{\varphi}_s$. For a perturbation h, $\hat{\varphi}_p$ and $\hat{\varphi}_s$ will have cancellation errors of $O(\epsilon/h)$. Thus, applying (13) to (20), the relative cancellation error in $\hat{\Phi}$ will be $O(\epsilon/h^2)$, assuming that $h \gg \epsilon/h$. Since the truncation error in Φ is $O(h)$, the optimal interval in this case is $\sqrt[3]{\epsilon}$. Figure 2 depicts the relevant six-digit mantissas when using $\hat{\Phi}$ to approximate f'' with $h = \sqrt[3]{\epsilon}$. In this case the two least significant digits of $\hat{\varphi}_p$ and $\hat{\varphi}_s$ will always be unreliable, and their computed difference will have an associated cancellation error of $\sqrt[3]{\epsilon}$.

The foregoing analysis will not apply if f' or f'' vanishes, or if the derivatives of f vary substantially in magnitude at the point of interest. If f' is zero and $|f''| \approx |f|$, a relative perturbation in x of $O(h)$ will produce a relative perturbation in f of $O(h^2)$ rather than $O(h)$, and consequently a perturbation of $O(\sqrt{\epsilon})$ will generally produce identical figures in *all* the digits of the mantissas of $f(x)$ and $f(x+h)$. A finite-difference estimate of f' is thus bound to be inaccurate in a relative sense when f' is small compared to f and f'', since either or both sources of error will dominate the magnitude of f', regardless of whether they are balanced or not. Nonetheless, we can still compute a satisfactory estimate of f'' with $h = \sqrt[3]{\epsilon}$, because $\hat{\varphi}_p$ and $\hat{\varphi}_s$ will retain some *absolute accuracy*.

4.4 Computing estimates of f' and f''.

The values of $f'(x)$ and $f''(x)$ are required if (28) is used to compute estimates of the optimal finite-difference interval. Since these values are assumed to be

unavailable (otherwise, finite-difference intervals would not be a source of concern), we must be able to compute meaningful approximations to f' and f'' with some initial interval h in order to estimate the desired interval $\hat{h}$. Note that it is unnecessary to compute f' or f'' to high accuracy, since an estimate of $\hat{h}$ with just one correct figure will suffice.

We hope to be able to use a single initial finite-difference interval h to compute both estimates in order to reduce the number of times that the function is evaluated. This interval and its associated cancellation error can then be used as the values that appear explicitly in the formula (28). Since two function evaluations are required to compute $\hat{\Phi}$, these values may also be used to compute a central-difference approximation ϕ_c to f', using (29).

The procedure that we suggest is based on choosing h to be large enough so that the cancellation error in $\hat{\Phi}$ is less than $1/\beta$, which implies that the estimate of f'' has approximately two figures free of computational error. The corresponding interval should give an adequate central difference estimate of f', using (29).

For a well-scaled function, the interval $\sqrt{\epsilon}(1 + |x|)$ will generally produce *no* correct digits in $\hat{\Phi}$, while providing the best value of ϕ_p. Hence, the initial finite-difference interval used to compute $\hat{\Phi}$ is given by

$$d = \beta\sqrt{\epsilon}(1 + |x|), \tag{30}$$

because for a well-scaled function we would expect the resulting $\hat{\Phi}(h)$ to contain approximately two digits free of computational error. However, bad scaling in f may cause (30) to be a poor choice for the finite-difference interval.

The algorithm that we propose for choosing an appropriate value of h is based on the observations in Section 4.1 concerning the relationship between $\hat{\eta}_p$, $\hat{\eta}_s$, and Υ. If (30) is too small, $\hat{\Phi}$ will tend to display excessive cancellation error for one of two reasons: $\hat{\eta}_p$ and $\hat{\eta}_s$ are too large, or ϕ_p and ϕ_s are too close. If the interval (30) is too large (which implies that the truncation error in $\hat{\Phi}$ may be unacceptably large), the cancellation error will tend to be quite small. In effect, the interval is chosen to be the smallest possible for which the cancellation error still allows some accuracy. If the truncation error for such an interval is unacceptable, no satisfactory interval exists.

In the implemented version of the following algorithm, $\rho = \beta$, and $K = \lceil \tau/2 \rceil$, where $\lceil a \rceil$ is the smallest integer greater than or equal to a. The formal statement of the algorithm is:

Algorithm 1 (*Estimating f' and f'' by finite differences*).

1. [Initialization.] Evaluate $f(x)$. Set $k \leftarrow 1$, $h \leftarrow \rho\sqrt{\epsilon}(1 + |x|)$.
2. Evaluate $f(x + h)$, $f(x - h)$, and the corresponding finite-difference approximations and estimates of cancellation error.

3. If $\max(\hat{\eta}_p, \hat{\eta}_s) < 1/\rho$ or $k = K$, go to step 5; otherwise, go to step 4.

4. [Increase h.] Set $k \leftarrow k+1$, $h \leftarrow \rho h$ and return to step 2.

5. [Either the cancellation error in f' is small enough or h has been increased K times.] Set $h_s \leftarrow h$. If $\Upsilon < 1/\rho^2$ and $k = 1$, go to step 8; otherwise, go to step 6.

6. If $\Upsilon < 1/\rho$, go to step 10. If $k \leq K$ and $\Upsilon \geq 1/\rho$, go to step 7. Otherwise, set $h \leftarrow h_s$ and go to step 10.

7. [Increase h in order to find a suitable estimate of f''.] Set $k \leftarrow k+1$ and $h \leftarrow \rho h$. Evaluate $f(x+h)$, $f(x-h)$, and the corresponding finite-difference approximations and estimates of cancellation error. Return to step 6.

8. [Decrease h in order to increase cancellation error in $\hat{\Phi}$.] Set $k \leftarrow k+1$ and $h \leftarrow h/\rho$. Evaluate $f(x+h)$, $f(x-h)$, and the corresponding finite-difference approximations and estimates of cancellation error. Go to step 9.

9. If $1/\rho \geq \Upsilon \geq 1/\rho^2$, go to step 10. If $k \leq K$, go to step 8. Otherwise, set $h \leftarrow h_s$ and go to step 10.

10. [Test size of second derivative.] If $\hat{\Phi} > h \min(\hat{\phi}_p, \hat{\phi}_s)$, the algorithm terminates.

11. [Estimate third derivative.] Evaluate $f(x+\rho h)$, and compute γ, an estimate of f''':

$$\gamma = \frac{6}{(\rho^3-\rho)h^3}\Big(f(x+\rho h) + \frac{\rho}{2}\big(f(x-h) - f(x+h)\big) + f(x) - \frac{1}{2}\rho^2\big(f(x+h) - 2f(x) + f(x-h)\big)\Big).$$

The algorithm then terminates. ∎

In practice, $f(x-h)$ is not evaluated in step 2 if $\hat{\eta}_p$ is too large.

To illustrate the behavior of Algorithm 1, we consider finding the derivatives corresponding to the first component of the multivariate function

$$\begin{aligned} F(x) = {} & (x_1 - 100)^2 + (x_2 - 1000)^2 + (x_3 + x_4 - 1000)^2 \\ & + (x_5 - 1000)^2 - (x_6 + x_7 - 200)^2 \\ & + 10^{-6}(x_1 + 2x_2 + 3x_3 + 4x_4 + 5x_5 + 6x_6 + 7x_7 - 1900)^2, \end{aligned} \tag{31}$$

at the point $(0, 0, 400, 100, 0, 0, 0)^T$. Table 3 gives the trial values of h generated by Algorithm 1, along with the corresponding estimates of cancellation error, $\hat{\phi}_p$, and $\hat{\Phi}$. The exact values of the first and second derivatives (rounded to six figures) are −199.730 and 1.99820. The final value of $\hat{\phi}_c$ is −199.730.

Table 8

Results of Algorithm 1 with $d = 16\sqrt{\epsilon} = .15625 \times 10^{-1}$

h	η_p	T	ϕ_p	ϕ
d	$.346099 \times 10^{0}$	$.622344 \times 10^{0}$	$-.256000 \times 10^{3}$	$-.499800 \times 10^{4}$
$16d$	$.406226 \times 10^{-1}$	$.674364 \times 10^{0}$	$-.200000 \times 10^{3}$	$-.160000 \times 10^{2}$
16^2d	$.269658 \times 10^{-2}$	$.661249 \times 10^{-1}$	$-.195750 \times 10^{3}$	$.193750 \times 10^{1}$
16^3d	$.243541 \times 10^{-3}$	$.501949 \times 10^{-3}$	$-.195797 \times 10^{3}$	$.199789 \times 10^{1}$

4.5 Numerical examples.

The success of the suggested procedures for obtaining sensible finite-difference intervals depends on two assumptions: that the linear relationship (26) in $1/h$ holds for cancellation error, and that relative truncation error is reliably estimated by formula (24), using the estimates of f' and f'' produced by Algorithm 1. In this section we give some numerical examples of the computation of estimates of optimal finite-difference intervals, using Algorithm 1 to obtain initial finite-difference estimates of f' and f'', followed by application of formula (28) to compute $\hat{h}$.

Tables 4 and 5 contain the results of estimating an absolute finite-difference interval for each gradient component $g_j(x)$, $j = 1, \ldots, n$, for two multivariate functions. In each table, the first column contains the index of the gradient component. The second column contains the final value of h, as obtained by Algorithm 1, and the third column contains $\hat{h}$, computed from (28). The fourth column contains the value of h^*, which satisfies the nonlinear equation

$$\eta_p(h^*) - \left| \frac{\phi_p(h^*) - f'(x)}{f'(x)} \right| = 0.$$

Thus, h^* is the finite-difference interval for which the cancellation error and the exact relative truncation error are "equal", where the tolerance that defines equality must be chosen based on the limited accuracy of the computed value of ϕ_p.

In Table 4, we give results for the function:

$$F(x) = (x_1 + 10x_2)^2 + 5(x_3 - x_4)^2 + (x_2 - 2x_3)^4 + 10(x_1 - x_4)^4, \tag{32}$$

at the point $x = (3, -1, 0, 1)^T$. The reliability of both underlying assumptions is shown by the similarity of columns 3 and 4. In addition, Table 4 displays the

Table 4

Optimal and Estimated Intervals for a Well-Scaled Problem

j	h	$\hat{h}$	h^*
1	$.3906 \times 10^{-3}$	$.9247 \times 10^{-3}$	$.1156 \times 10^{-2}$
2	$.1953 \times 10^{-3}$	$.1405 \times 10^{-3}$	$.1440 \times 10^{-2}$
3	$.1562 \times 10^{-1}$	$.3079 \times 10^{-3}$	$.2560 \times 10^{-2}$
4	$.1953 \times 10^{-3}$	$.9214 \times 10^{-3}$	$.9214 \times 10^{-3}$

well-scaled nature of the function (32), since the optimal interval for the i-th variable is of order $\sqrt{\epsilon}(1 + |x_i|)$.

Table 5 contains the analogous results for the seven-variable badly scaled function (31). The optimal intervals are again closely predicted using the procedures of Sections 4.3 and 4.4.

5. Using finite differences in minimization

A descent method for minimising $F(x)$, $x \in \mathbb{R}^n$, proceeds as follows. Let k denote the current iteration, and let x_k denote the k-th estimate of the solution (starting with $k = 0$ at the initial point x_0). Each iteration requires $g(x_k)$ (sometimes denoted by g_k), the gradient vector $\nabla F(x)$ evaluated at x_k. At the k-th iteration a direction of search p_k is computed such that the directional derivative $g_k^T p_k$ is negative, and the new estimate x_{k+1} is given by $x_k + \alpha_k p_k$, where α_k (a positive *step length*) is chosen such that $F(x_k + \alpha_k p_k) < F(x_k)$.

Quasi-Newton methods are probably the most commonly used techniques for unconstrained minimisation when only first derivatives are available (see Dennis and Moré, 1977). In a quasi-Newton method, the direction of search is defined

Table 5

Optimal and Estimated Intervals for a Badly Scaled Problem

j	h	$\hat{h}$	h^*
1	$.6400 \times 10^{2}$	$.1773 \times 10^{1}$	$.1773 \times 10^{1}$
2	$.6400 \times 10^{2}$	$.1482 \times 10^{1}$	$.1758 \times 10^{1}$
3	$.6266 \times 10^{1}$	$.1494 \times 10^{1}$	$.1724 \times 10^{1}$
4	$.2525 \times 10^{2}$	$.1486 \times 10^{1}$	$.1735 \times 10^{1}$
5	$.6400 \times 10^{2}$	$.1496 \times 10^{1}$	$.1740 \times 10^{1}$
6	$.4000 \times 10^{1}$	$.1442 \times 10^{1}$	$.1285 \times 10^{1}$
7	$.4000 \times 10^{1}$	$.1401 \times 10^{1}$	$.1250 \times 10^{1}$

as the solution of the linear system

$$B_k p_k = -g_k, \tag{33}$$

where B_k is an approximation to the Hessian matrix. After each iteration the approximate Hessian is updated, the most popular update being the BFGS formula

$$B_{k+1} = B_k + \frac{1}{g_k^T p_k} g_k g_k^T + \frac{1}{\alpha_k y_k^T p_k} y_k y_k^T,$$

where $y_k = g_{k+1} - g_k$ (see Dennis and Moré, 1977). The BFGS update is a special case of the more general Broyden one-parameter family

$$B_{k+1}^{\phi} = B_k^{\phi} + \frac{1}{g_k^T p_k} g_k g_k^T + \frac{1}{\alpha_k y_k^T p_k} y_k y_k^T - \phi_k g_k^T p_k w_k^T w_k, \tag{34}$$

where

$$w_k = \frac{1}{y_k^T p_k} y_k - \frac{1}{g_k^T p_k} g_k$$

and ϕ_k is a scalar function of y_k and $\alpha_k p_k$ (see Broyden, 1970; Gill and Murray, 1972).

A quasi-Newton method can be implemented even when first derivatives are not available by using a finite-difference estimate $\hat{g}_k$ of the gradient vector g_k. We now consider two methods for choosing a reasonable finite-difference interval to compute the approximate gradient.

5.1 Stewart's modification of a quasi-Newton method.

Stewart (1967) suggested a modification of a quasi-Newton method in which the finite-difference interval is re-estimated at each iteration. In the following discussion, we suppose that the k-th iteration has been performed, and that $\hat{g}_k$ is available. Let γ_j denote the jth element of $\hat{g}_k$, and Γ_j the jth diagonal element of B_k.

The finite-difference interval is chosen by balancing estimates of relative truncation and cancellation errors. Both sources of error are estimated using a local quadratic approximation based on the function value at x_{k+1}, and the approximate gradient and quasi-Newton approximation to the Hessian from the previous iteration:

$$F(x_{k+1} + h_j e_j) \approx q(h_j) = F_{k+1} + h_j \gamma_j + \frac{1}{2} h_j^2 \Gamma_j.$$

The relationship that balances the two sources of error is then

$$\left|\frac{2\epsilon F_{k+1}}{h_j\gamma_j + \frac{1}{2}h_j^2\Gamma_j}\right| = \frac{h_j}{2}\left|\frac{\Gamma_j}{\gamma_j}\right|, \tag{35}$$

where the left-hand side of (35) represents the relative truncation error, and the right-hand side is an estimate of the relative cancellation error. The desired value of h_j then solves the resulting cubic equation.

It is customary in quasi-Newton methods to recur either the inverse or a factorization of B_k, in order to facilitate the solution of equation (33). Consequently, with Stewart's approach the diagonal elements of B_k must be recurred separately. To recur these elements for the BFGS update, for example, let γ_j and ψ_j denote the j-th elements of $\hat{g}_k$ and $\hat{y}_k$ respectively, where $\hat{y}_k$ is computed with the approximate gradients; let $\bar{\Gamma}_j$ denote the j-th diagonal of the approximate Hessian during iteration $k+1$. Then

$$\bar{\Gamma}_j = \Gamma_j + \frac{1}{g_k^T p_k}\gamma_j^2 + \frac{1}{\alpha_k y_k^T p_k}\psi_j^2.$$

Stewart's estimate of the finite-difference interval depends critically upon the assumption that the diagonal elements of the approximate Hessian matrix are similar in *magnitude* to the diagonal elements of the true Hessian. The following theorem indicates that this assumption is unjustified, even after the first n iterations. It is unlikely to be true before this point because the set of generated directions does not span the full space.

Theorem 1. *Let $F(x)$ be any twice-continuously differentiable function. Let $\{x_k\}$ be a sequence of points generated by applying the BFGS update, for some given initial x_0 and Hessian approximation B_0, where each α_k is the smallest positive step from x_k to a minimum of F along p_k. Let $\{x_k^\phi\}$ be the sequence generated by using update (34), with the same starting point and initial approximate Hessian. Let p_k^ϕ, α_k^ϕ and B_k^ϕ denote the corresponding values of p_k, α_k and B_k respectively, where α_k^ϕ satisfies the same criteria as α_k. If each of the sequences $\{B_k\}$ and $\{B_k^\phi\}$ is well-defined, then*

$$B_k = B_k^\phi + \left(\frac{\phi_{k-1}}{g_{k-1}^T p_{k-1}^\phi}\right) g_k g_k^T, \tag{36}$$

and

$$\alpha_k p_k = \alpha_k^\phi p_k^\phi. \tag{37}$$

Proof. A proof is given by Dixon (1972). ∎

Equation (37) implies that the same sequence $\{x_k\}$ will be generated by any well-defined member of the Broyden class of quasi-Newton updates when the step length at each iteration is the nearest minimum along the direction of search. Similarly, (36) implies that all the approximate Hessian matrices generated by the one-parameter family differ by a matrix of rank one. Thus, *even for a quadratic function*, the elements of the approximate Hessian will generally be poor approximations to the elements of the true Hessian.

One unfortunate consequence of this result is that for functions that are well scaled, the interval found by Stewart's method may differ substantially from $\sqrt{\epsilon}$, and hence the gradient elements will contain unnecessary error.

5.2 An alternative proposal.

We propose an alternative procedure for selecting the finite-difference interval, based on the following observations. The procedures described in Section 4 require at least $2n$ function evaluations to compute a satisfactory set of intervals, and thus would be prohibitively expensive if carried out at every iteration. Although Stewart's method does not require additional function evaluations, we do not favor its use because there is no theoretical basis for the associated estimates of truncation and cancellation errors, as shown in Section 5.1. Furthermore, in our experience the estimates produced by Stewart's method do not reliably reflect the true size of the errors.

The proposed method executes the procedures of Section 4 at the initial point to develop a set of absolute intervals $\hat{h}_j$, $j = 1, \ldots, n$. The relative interval δ_j used for the j-th variable is

$$\delta_j = \begin{cases} \hat{h}_j, & |x_j| \leq 1, \\ \hat{h}_j/|x_j|, & |x_j| > 1, \end{cases}$$

where x is the point at which $\{\hat{h}_j\}$ are computed. In effect, $\hat{h}_j$ is interpreted as an absolute step when $|x_j|$ is of order unity or less, and as a *relative* perturbation otherwise.

The approximate gradients are computed with the forward-difference formula (2) until they appear to have become "unreliable". For example, the forward-difference estimate is not used when the relative cancellation error in every component of the approximate gradient exceeds $1/\beta$. In this case, a switch is made to the central-difference approximation (3), with the slightly larger interval

$$\bar{\delta}_j = \frac{\delta_j}{\epsilon^{1/6}},$$

which is derived from an analysis similar to that in Section 4.3. To conserve the number of evaluations of $F(x)$, it is recommended that the first central-

difference approximation be computed with h_j, since the set of function values $\{F(x + h_j e_j)\}$ will have already been computed for the forward-difference estimate. A switch to central differences is also made if the step length α_k is such that the change in x will be less than the current finite-difference interval. The reader is referred to Gill and Murray (1972) for further details concerning switches in formulae.

An advantage of carrying out the estimation of the optimal h at the starting point is that an initial central-difference approximation to the gradient is thereby produced, as well as estimates of the diagonal elements of the Hessian matrix, which may be used to initialize the Hessian approximation for a quasi-Newton method. Retaining the same finite-difference interval throughout the minimization results in similar truncation errors in g_k and g_{k+1}, which is beneficial when computing the vector y_k needed to perform the quasi-Newton update. This latter property implies that the finite-difference version of any quasi-Newton method that displays n-step termination with exact gradients will also achieve n-step termination when F is a well-scaled quadratic function.

A word of warning is appropriate at this point. There may not exist a fixed set of intervals that are appropriate throughout the range of points where gradients must be approximated during a minimization, and consequently, the given procedure may not produce a sensible set of intervals for every point subsequently generated. Nonetheless, in most practical applications it is adequate to examine properties of the function in some detail at a point that typifies those for which the gradient will be required. In the unlikely event that a minimization algorithm fails because of a poor choice of finite-difference interval, it can be restarted in order to produce intervals more suitable for the neighbourhood in which failure occurred.

6. Conclusions

We have described methods for estimating reasonable finite-difference intervals for several finite-difference formulae. These techniques are useful not only in optimization algorithms, but in other circumstances (such as sensitivity analysis) where accurate approximations to derivatives are required. Automatic scaling procedures (see, e.g., Gill *et al.*, 1980) also rely on the use of finite-difference approximations to analyse the behavior of functions.

The cost of implementing the method of Section 5.2 (except for badly scaled functions) is typically only $2n$ function evaluations, which are carried out only at the initial point. This number of additional evaluations is unlikely to be significant, particularly in view of the improved efficiency during the optimization; for example, delaying the switch to central differences by just one iteration will tend to result in a net gain in efficiency. For badly scaled functions, the

use of a well-chosen finite-difference interval may be crucial to the success of an algorithm.

In this paper, we have been concerned solely with the unconstrained problem; however, a similar analysis can (and should) be extended to constrained problems. It is sometimes thought that the constrained case is easier because the gradient of the objective function is not necessarily zero at the solution. Nonetheless, some projection of the gradient does decrease to zero, and the accuracy of any approximation to the projected gradient is critical in computing the search direction. In fact, in a constrained problem where derivatives are not available, it is usually more efficient, and certainly superior in terms of numerical stability, to approximate the projected gradient directly by finite differences along selected vectors. In this case, it is clear that the selection of a reasonable finite-difference interval is just as important as in the unconstrained case.

References

Broyden, C. G. (1970). The convergence of a class of double-rank minimisation algorithms, *J. Inst. Maths Applics.*, **6**, pp. 76–90.

Dennis, J. E. and Moré, J. J. (1977). Quasi-Newton methods, motivation and theory, *SIAM Review*, **19**, pp. 46–89.

Dixon, L. C. W. (1972). Quasi-Newton algorithms generate identical points. II. The proof of four new theorems, *Math. Prog.*, **3**, pp. 345–358.

Gill, P. E. and Murray, W. (1972). Quasi-Newton methods for unconstrained optimisation, *J. Inst. Maths. Applics.*, **9**, pp. 91–108.

Gill, P.E., Murray, W., Saunders, M. A., and Wright, M. H. (1980). Automatic scaling algorithms for nonlinear optimisation, to appear.

Kahan, W. (1973). The implementation of algorithms: Part 1, Technical Report 20. Department of Computer Science, University of California, Berkeley.

Stewart, G. W. (1967). A modification of Davidon's method to accept difference approximations of derivatives, *J. A. C. M.*, **14**, pp. 72–83.

Wilkinson, J. H. (1963). *Rounding Errors in Algebraic Processes*, Notes on Applied Sciences No. 32, Her Majesty's Stationary Office, London; Prentice-Hall, New Jersey.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER SOL 80-6	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Computing the Finite-Difference Approximations to Derivatives for Numerical Optimization		5. TYPE OF REPORT & PERIOD COVERED TECHNICAL REPORT 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Philip E. Gill, Walter Murray, Michael A. Saunders, and Margaret H. Wright		8. CONTRACT OR GRANT NUMBER(s) DAAG29-79-C-0110
9. PERFORMING ORGANIZATION NAME AND ADDRESS Department of Operations Research - SOL Stanford University Stanford, CA 94305		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS U.S. Army Research Office P.O. Box 12211 Research Triangle Park, NC 27709		12. REPORT DATE May 1980 13. NUMBER OF PAGES 22
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

This document has been approved for public release and sale; its distribution in unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

THE VIEW, OPINIONS, AND/OR FINDINGS CONTAINED IN THIS REPORT ARE THOSE OF THE AUTHOR(S) AND SHOULD NOT BE CONSTRUED AS AN OFFICIAL DEPARTMENT OF THE ARMY POSITION, POLICY, OR DECISION, UNLESS SO DESIGNATED BY OTHER DOCUMENTATION.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

finite-difference approximations
truncation error
cancellation
optimization without derivatives
floating-point arithmetic.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

SEE ATTACHED

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

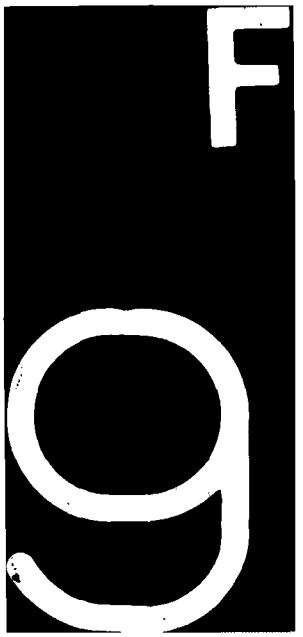
UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

SOL 80-6 COMPUTING THE FINITE-DIFFERENCE APPROXIMATIONS TO DERIVATIVES FOR NUMERICAL OPTIMIZATION

by Philip E. Gill, Walter Murray, Michael A. Saunders, and Margaret H. Wright

Finite-difference approximations to derivatives are useful not only in optimization algorithms, but also in other circumstances such as sensitivity analysis. In this paper we discuss methods for estimating the relative cancellation error and relative truncation error in a finite-difference approximation and propose a technique for computing the finite-difference interval so that the bounds upon the errors are balanced. We also propose a method for choosing the finite-difference interval in a quasi-Newton algorithm for unconstrained minimization that uses function values only.


F
9